(12) UK Patent Application (19) GB (11) 2 334 354 (13) A

(43) Date of A Publication 18.08.1999

(21) Application No 9802974.7

(22) Date of Filing 13.02.1998

(71) Applicant(s)
International Business Machines Corporation
(Incorporated in USA - New York)
Armonk, New York 10504, United States of America

(72) Inventor(s)
Colin David McCall
Jane Henderson Shaw

(74) Agent and/or Address for Service
C Boyce
IBM United Kingdom Limited, Intellectual Property Department, Mail Point 110, Hursley Park, WINCHESTER, Hampshire, SO21 2JN, United Kingdom

(51) INT CL[6]
G06F 9/445

(52) UK CL (Edition Q)
G4A AFL

(56) Documents Cited
GB 2311389 A EP 0358292 A2

(58) Field of Search
UK CL (Edition P) G4A AFL
INT CL[6] G06F 9/445

(54) Abstract Title
Booting a computer system from a network

(57) A client computer system 10 is adapted to connect to a server 16 across a network 20. The client system includes memory (25 or a storage medium) into which an operating system is loadable. The client includes a network card 12 having a PROM 14 adapted to download a bootstrap program 17 from the server. The bootstrap program is, in turn, adapted to load an operating system image 18 over the network from the server into a location in the memory. The client then boots from the operating system image in memory, with the client system being adapted subsequently to modify the operating system image in memory, without reloading the image over the network, and to re-boot the client system.

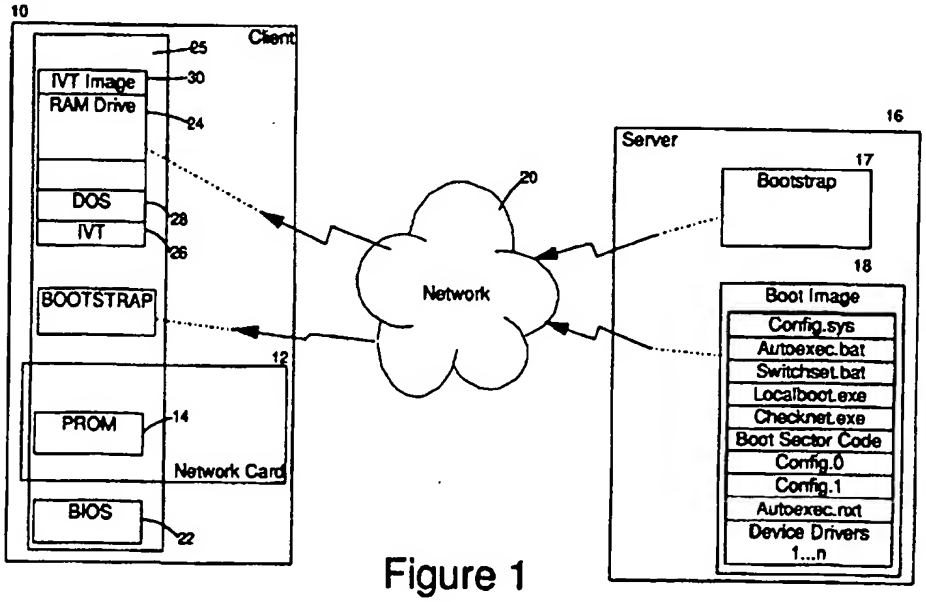

Figure 1

BEST AVAILABLE COPY

GB 2 334 354 A

At least one drawing originally filed was informal and the print reproduced here is taken from a later filed formal copy.

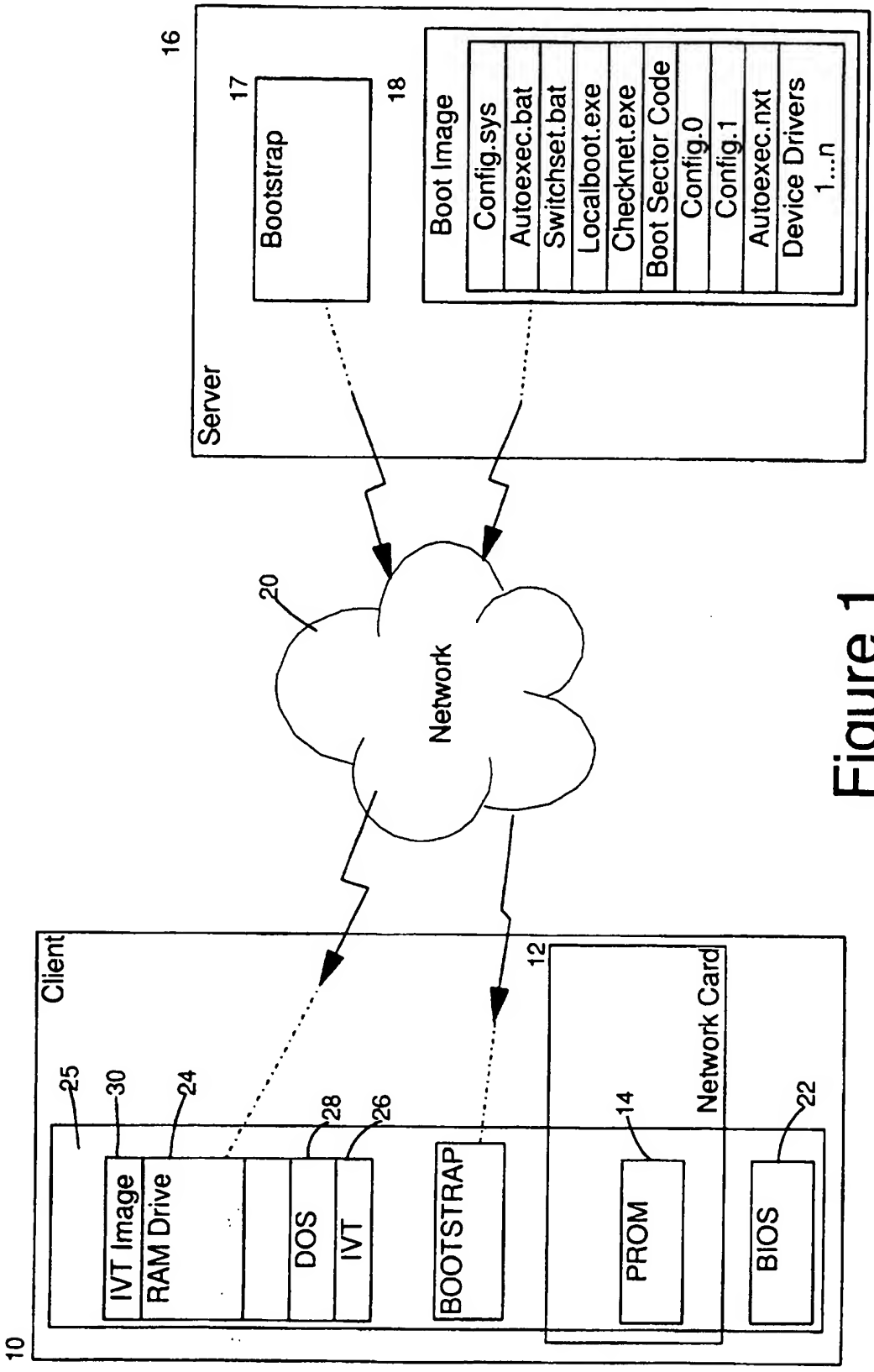

Figure 1

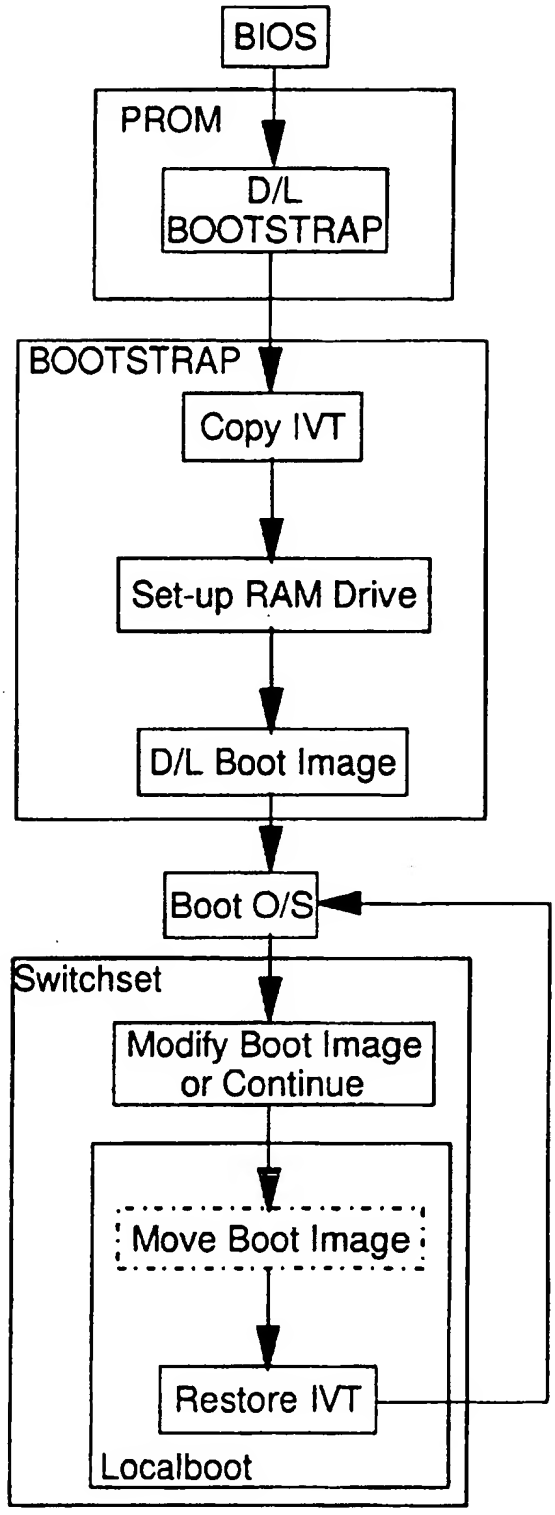

Figure 2

2334354

BOOTING A COMPUTER SYSTEM FROM A NETWORK

The present invention relates to booting a computer system from a network.

It is known for a client computer system, connected via any one of a number of commercially available network media to a server, to download a boot image from the server and boot. For example, IBMs LCCM Version 2 product, in common with many other current and imminent personal computer (PC) management and maintenance tools such as Intel's LANDesk Configuration Manager, uses the dynamic host configuration protocol (DHCP) and trivial file transfer protocol (TFTP) to load a DOS operating system as a diskette image into a virtual RAM drive in a client system's memory, and then boots from the diskette image, to perform installation and maintenance tasks.

Different client computer systems booting up using different boot images downloaded from a network require different boot images depending on the hardware configuration of the client and the function to be performed by the booted client. This requires the server to be aware of each client computer configuration and function to be performed when configuring the boot image.

Also, some tasks require that the client be re-booted one or more times in succession to perform a series of functions before proceeding. For example to partition and format a hard disk under DOS, the client must be rebooted between the partition and format stages. A second network boot entails a significant delay as well as creating additional network traffic, and requires that the server computer configuration is modified between the first and second boot stages so the client computer boots with the correct software for each stage in the correct sequence.

The Preboot Execution Environment (PXE) specification (Intel, Compaq et al) allows different network cards to operate with the same OS image, thus solving the major problem of ensuring compatibility between the booted image and the system network card.

However, many older PCs and network adapters do not implement the PXE specification, and other compatibility problems remain requiring different versions of the operating system image to be downloaded for different client systems or tasks. For example, some maintenance tasks

require the presence of an Extended Memory Manager (like EMM386) while others will not operate with an Extended Memory Manager. This requires the server to be re-configured to provide the correct boot image for the task to be performed.

It is an object of the present invention to mitigate these problems.

Accordingly, the present invention provides a client system as claimed in claim 1.

In a further aspect the invention provides a method of booting a computer as claimed in claim 12.

In a still further aspect the invention provides an operating system images as claimed in claim 14.

Embodiments of the invention will now be described with reference to the accompanying drawings, in which:

Figure 1 is a schematic diagram of a network computer system including a host computer system and a client system according to a preferred embodiment of the invention; and

Figure 2 is a flow diagram of the operation of the preferred embodiment.

In the present invention, as with conventional network booting, a client system 10 includes a network card 12 including a PROM 14 programmed to connect the client to a server 16 and to download a bootstrap program 17 from the server via the network 20. The bootstrap 17 in turn is programmed to download a boot image 18 from the server via the network 20. The PROM program runs immediately after a BIOS or initial bootstrap program 22 and before an operating system is loaded. In the case of the DOS operating system, the boot image includes a number of files, for example, autoexec.bat, config.sys and any other driver files necessary to boot the client.

In the present embodiment, the bootstrap 17 is programmed to set up a RAM drive 24 in memory 25 into which the boot image 18 files are transferred under the control of the PROM. The client may alternatively

include a floppy disk drive (not shown) and the boot image 18 could be transferred onto the floppy disk avoiding the necessity for a RAM drive. This, however, would prove a slower and more cumbersome solution although not outside the scope of the present invention.

The DOS operating system employs an interrupt vector table (IVT) 26 which contains pointers to interrupt service routines normally located within the DOS portion 28 of memory. When DOS is loaded into memory, however, it affects the initial state of the interrupt vector table, pointing the IVT entries to the location of interrupt service routines in memory. If the client operating system is to be re-booted, then the state of the interrupt vector table before DOS is loaded needs to be preserved.

Thus, in the present embodiment, the bootstrap program preserves the interrupt vector state by copying the interrupt vector table 26 contents located in memory at address 0 to 400h to a location 30 alongside the RAM drive 24 for the boot image. When DOS starts, the boot image is modified as required, Figure 2. A program "localboot", explained later, then restores the interrupt vector state from the copy 30 in RAM, before returning to reboot from the boot image stored in the RAM drive 24.

The program 'localboot' first performs a memory copy of the interrupt vector table from the location in RAM where it has been stored back to location 0h, then performs a memory copy of the boot image boot sector code from the RAM drive to location 7c00h in memory, then passes control to the boot sector with a jump instruction to location 7c00h, emulating the standard boot process for a PC.

The boot image 18 is generic and is used for all clients and all tasks. In one example, the generic boot image contains a batch file, "Switchset.bat", started from "autoexec.bat", which, if required, modifies the image in the RAM drive 24 according to the requirements of the client configuration. In the example, 2 files, config.sys and protocol.ini, are modified before a second boot by copying different versions of batch file, corresponding to different types of network adapter, depending on the network card installed. "Switchset" then forces the client to reboot, with the command "localboot", from the modified boot image in the RAM drive.

In more detail, the original config.sys does not load any network device drivers, since the first boot is applicable to multiple network card types:

Config.sys

```
DOS=HIGH,UMB
FILES=30
STACKS=9,216
DEVICE=A:\DOS\HIMEM.SYS
LASTDRIVE=Z
```

In the present embodiment, the original autoexec.bat file just calls the Switchset batch file to do the work:

Autoexec.bat

```
SWITCHSET
```

Switchset.bat, listed below, determines which of two network drivers is required based on a program, "Checknet", and sets up the appropriate config.sys and protocol.ini files for the required drivers. Switchset also copies autoexec.nxt to autoexec.bat to set up the process to be executed on the next boot, then runs "localboot" which restores the interrupt vector environment from memory and reboots from the modified copy of the operating system in the RAM drive 24.

Switchset.bat

```
CALL CHECKNET
IF ERRORLEVEL 1 GOTO TYPE1

:TYPE0
COPY CONFIG.0 CONFIG.SYS /Y
COPY IBMNET\PROTOCOL.0 IBMNET\PROTOCOL.INI /Y
GOTO REBOOT

:TYPE1
COPY CONFIG.1 CONFIG.SYS /Y
COPY IBMNET\PROTOCOL.1 IBMNET\PROTOCOL.INI /Y

:REBOOT
COPY AUTOEXEC.NXT AUTOEXEC.BAT /Y
localboot
```

CONFIG.0 is a version of config.sys which loads a generic network adapter driver A:\DOS\NDIS.DOS and uses a utility PXUTIL.SYS to modify PROTOCOL.INI, NETWORK,INI and AUTOEXEC.BAT based on data passed in the network boot protocol-

Config.0

```
DOS=HIGH,UMB
FILES=30
STACKS=9,216
DEVICE=A:\DOS\HIMEM.SYS
LASTDRIVE=Z
DEVICE=A:\IBMNET\PXPUTIL.SYS -a A:\IBMNET\PROTOCOL.INI
DEVICE=A:\IBMNET\PXPUTIL.SYS -a A:\IBMNET\NETWORK.INI
```

```
DEVICE=A:\IBMNET\PXPUTIL.SYS -y 3 -a A:\AUTOEXEC.BAT
DEVICE=A:\IBMNET\PXPUTIL.SYS -a A:\AUTOEXEC.BAT
DEVICE=A:\IBMNET\PROTMAN.DOS /I:A:\IBMNET
DEVICE=A:\DOS\NDIS.DOS
DEVICE=A:\IBMNET\NTSTS.DOS
DEVICE=A:\IBMNET\DLSHELP.SYS
```

CONFIG.1 is a version of config.sys which loads a specific network adapter driver A:\DOS\ENDS2ISA.DOS for a specific network card that does not support the generic adapter. The PXUTIL utility is replaced by another utility BPUTIL which works with the specific adapter.

Config.1

```
DOS=HIGH,UMB
FILES=30
STACKS=9,216
DEVICE=A:\DOS\HIMEM.SYS
LASTDRIVE=Z
DEVICE=A:\IBMNET\BPUTIL.SYS -a A:\IBMNET\PROTOCOL.INI
DEVICE=A:\IBMNET\BPUTIL.SYS -a A:\IBMNET\NETWORK.INI
DEVICE=A:\IBMNET\BPUTIL.SYS -y 3 -a A:\AUTOEXEC.BAT
DEVICE=A:\IBMNET\BPUTIL.SYS -a A:\AUTOEXEC.BAT
DEVICE=A:\IBMNET\PROTMAN.DOS /I:A:\IBMNET
DEVICE=A:\DOS\ENDS2ISA.DOS
DEVICE=A:\IBMNET\NTSTS.DOS
DEVICE=A:\IBMNET\DLSHELP.SYS
```

PROTOCOL.0 and PROTOCOL.1 contain corresponding variations in the network configuration.

After switchset calls localboot and the interrupt vector table has been restored, the client can be re-booted. On the second boot, either config.0 or config.1, now config.sys, is loaded and the previous autoexec.nxt, now autoexec.bat, shown below, connects to the networked server 16 which is now accessible since the correct network drivers have been installed.

autoexec.nxt

```
\IBMNET\net logon %CLIENT_NAME% /pwcaching:no
\ibmnet\net use s: \\#@t128*##########\lanc$$
\ibmnet\net use t: \\#@t128*##########\lccm$tmp
s:\_lccmD.bat
```

The batch file _LCCMD.BAT, called from autoexec.bat, resides on the server and contains instructions for executing the specific task required. Based on these instructions, the server 16 may further modify the boot image in the RAM drive 24 to support the required task, for example by modifying the config.sys file to load or not load EMM386 on the next boot, and boot again by calling "localboot".

This sequence may be repeated an arbitrary number of times to perform a sequence of processes without ever rebooting from the network, or requiring the server to modify the generic boot image stored on the server.

A problem with the RAM drive embodiment of a network boot is that when the PROM 14 program loads the bootable image 18 into the RAM drive 24, it must choose a fixed location in memory to locate the RAM drive, or calculate a location without any awareness of the task that is to be performed. Some tasks may also need to use the same fixed location, overwriting and corrupting the boot image. For example, a program to upgrade the client systems's BIOS code 22 in flash memory may assume that no other program is running and choose an arbitrary area in memory to store its data. In this case, the generic boot image may both modify itself and copy itself to a different part of RAM before rebooting. Conventional programs are available to achieve this, or localboot can be adapted to perform a byte-for-byte memory copy from the current location of the bootable image in RAM to another location.

It will be seen that in a variation of the present embodiment, the operating system image 18 could be divided into a generic portion and one or more optional portions. The generic portion could be downloaded first and one of the or each optional portions could be downloaded, if necessary, after a second or subsequent boot of the operating system. It will be seen, however, that in any case it is not necessary to download the generic portion of the operating system more than once.

CLAIMS

1. A client computer system adapted to connect to a server across a network, the client system including memory into which an operating system is loadable and being adapted to load at least a portion of an operating system image over the network from the server into a location in said memory and boot from the operating system image in memory, wherein said client system is adapted to subsequently modify the operating system image in memory without reloading the portion of the operating system image over the network and re-boot the client system.

2. A client system as claimed in claim 1 wherein the client system is adapted to modify said location of the operating system image in memory before rebooting.

3. A client system as claimed in claim 1 wherein the client system is adapted to modify said operating system image and reboot several times for complex sequences of functions requiring different operating system images.

4. A client system as claimed in claim 1, said system including a network card, said network card including a programmed memory operable to download a bootstrap program from said server.

5. A client system as claimed in claim 4, wherein said bootstrap program is operable set up a virtual drive in memory and to load said operating system image into said virtual drive over the network.

6. A client system as claimed in claim 5 wherein said operating system is the DOS operating system, and said operating system image includes a plurality of files, said files including one or more of an autoexec.bat, a config.sys or a device driver file.

7. A client system as claimed in claim 6 in which said bootstrap program is operable to duplicate the contents of the DOS interrupt vector table, prior to starting DOS.

8. A client system as claimed in claim 7 in which said operating system image includes a program operable to restore the contents of the DOS interrupt vector table from said duplicate, prior to re-starting DOS.

9. A client system as claimed in claim 8 wherein said program is operable to relocate said operating system image in memory.

10. A client system as claimed in claim 1 wherein said operating system image is downloaded into volatile memory.

11. A client system as claimed in claim 1 comprising a storage medium and wherein said operating system image is downloaded onto said storage medium.

12. A method of booting a computer in a network said method including the steps of:

a. downloading at least a portion of an operating system image from a network server

b. booting the operating system

c. modifying the operating system image in memory

d. repeating steps b. and c.

13. A method of booting a computer as claimed in claim 12 comprising the steps of downloading a bootstrap program from a network server, said bootstrap program being operable to download said operating system image from the network server.

14. An operating system image for a client computer system including a network card operable to connect said client system to a server across a network, said operating system image being operable to boot said client system, to modify itself after being booted and to re-boot said client system to connect said client system to said server.

15. An operating system image as claimed in claim 14 comprising a plurality of DOS operating system files, said files including a program adapted to check the state of said client system and to modify one or more of a config.sys and/or an autoexec.bat file accordingly.

16. An operating system image as claimed in claim 15, said files comprising a program adapted to re-boot said client system.

The Patent Office

Application No:	GB 9802974.7	**Examiner:**	Leslie Middleton
Claims searched:	1,12,14	**Date of search:**	11 August 1998

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.P): G4A (AFL)

Int Cl (Ed.6): G06F 9/445

Other: Online: INSPEC, WPI.

Documents considered to be relevant:

Category	Identity of document and relevant passage		Relevant to claims
A	GB 2311389 A	(International Business Machines Corporation)	
A	EP 0358292 A2	(Digital Equipment Corporation)	

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.